Synthetic Tait Computability for Simplicial Type Theory

Jonathan Weinberger, Benedikt Ahrens, Ulrik Buchholtz, Paige North

_, Delft University of Technology, University of Nottingham, University of Pennsylvania

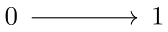
TYPES 2022
Nantes/The Internet
July 1, 2022

A type theory for synthetic $(\infty, 1)$-categories

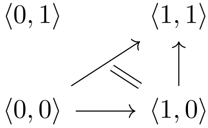
- Homotopy type theory (HoTT) is a synthetic theory of ∞**-groupoids**, via the (undirected) path types $(a =_A b)$.
- But for $(\infty, 1)$**-categories** we need **directed** arrow types $(a \to_A b)$.
- Riehl–Shulman '17: synthetic $(\infty, 1)$-category theory in a **simplicial** extension of HoTT
- Further developed by Cavallo–Riehl–Sattler '18, Buchholtz–W '21, W '21 & '22, Bardomiano Martínez '22; and in a **bicubical setting** by Weaver–Licata '20 & '21
- Building on Riehl–Verity's **model independent** ∞**-category theory** '22
- Prototype proof assistant for STT by Nikolai Kudasov: `rzk`

STT: Examples of shapes

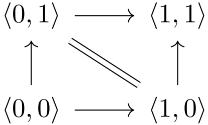
Δ^1



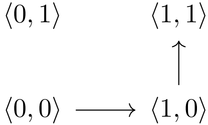
Δ^2



$\Delta^1 \times \Delta^1$



Λ^2_1



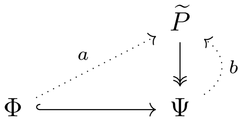
$$\Delta^1 :\equiv \{t : \mathbb{2} \mid \top\}, \quad \Delta^2 :\equiv \{\langle t, s\rangle : \mathbb{2} \times \mathbb{2} \mid s \leq t\},$$
$$\Delta^1 \times \Delta^1 \equiv \{\langle t, s\rangle : \mathbb{2} \times \mathbb{2} \mid \top\}, \quad \Lambda^2_1 :\equiv \{\langle t, s\rangle : \mathbb{2} \times \mathbb{2} \mid (s \equiv 0) \vee (t \equiv 1)\}$$

STT: Extension types

Idea: "Π-types with strict side conditions". Originally due to Lumsdaine–Shulman.[1]

Input:

- shape inclusion $\Phi \hookrightarrow \Psi$
- family $P : \Psi \to \mathcal{U}$
- partial section $a : \Pi_{t:\Phi} P(t)$

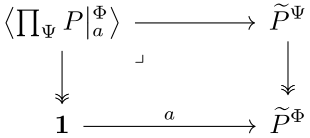


$\rightsquigarrow$

Extension type $\left\langle \prod_\Psi P \big|^{\Phi}_{a} \right\rangle$

with terms $b : \Pi_\Psi P$ such that $b|_\Phi \equiv a$

Semantically:



Can now define *e.g.* hom-types
$(x \to_A y) \equiv \left\langle \Delta^1 \to A \Big|^{\partial \Delta^1}_{[x,y]} \right\rangle$.

[1] cf. also path types in Cubical Type Theory

Our work: Metatheory of STT

- *Goal:* Provide an appropriate logical framework for STT to prove metatheorems such as normalization.
- *Method:* Adapt the setup of **synthetic Tait computability (STC)** by Sterling '21 and Sterling–Angiuli '21 and their work on cubical type theory.
- *Construction:* Relative to an abstract logical framework, build a **normalization topos** for the respective type theory (ETT, CTT, STT, . . .) and carry out a version of classical **normalization by evaluation (NBE)** in the internal language.[2]
- *Progress so far:* Adapted and adjusted many key components of [Ste21] and [SA21] to simplicial type theory.
- Ultimately, this should yield—in a very analogous way—the same meta-theoretic results for STT as Sterling and Sterling–Angiuli proved for CTT, such as:

  Normalization, idempotence of normalization, and decidability of judgmental equality

[2] a modal extensional type theory capturing both syntactic and semantic data

Further work

- Traditional **Tait computability** goes back to Tait '67, later refined by Girard and Martin-Löf.
- Internalizes and/or generalizes earlier work by Altenkirch–Hofmann–Streicher '95, Čubrić–Dybjer–Scott '98, Fiore '02, Abel '13, Altenkirch–Kaposi '16
- Alternative account to metatheoretic results about CTT by Huber '18, Coquand '18, Angiuli–Favonia–Harper '18, Coquand–Huber–Sattler '19, Uemura '21, and Kolomatskaia '22 (`stlc` formalization).
- STC has furthermore been used in works[3] by Sterling '21 & '22, Sterling–Angiuli '21, Gratzer–Birkedal '22, Gratzer '21 & '22, Niu–Sterling–Grodin–Harper '22, Sterling–Harper '21$^{\times 3}$, and Uemura '22.

[3] See Sterling's online `STC bibliography`.

Towards normalization of STT

Theorem

There is a computable function assigning to every type

$$\Xi \mid \Phi \mid \Gamma \vdash A$$

and every term

$$\Xi \mid \Phi \mid \Gamma \vdash a : A$$

in simplicial type theory a unique normal form.

Normalization, Step 1: Present STT via abstract syntax

- Idea (Sterling): To simplify and conceptualize the metatheory, present type theories in an appropriate LF through “abstract signatures” (*objective syntax*) rather than by raw terms/concrete syntax.
- Ingredients: judgments for cubes $\mathbb{C}$, topes $\mathbb{T}$, and types ty
- Families over simplicial shapes:

$$(A \twoheadrightarrow \Psi \rightarrowtail I) \leftrightsquigarrow \left\langle I : \mathbb{C}, \Psi : [I] \to \mathbb{T}, A : \left(\sum_{i:[I]} [\Psi(i)] \right) \to \mathrm{ty} \right\rangle$$

- Extension types:

$$\left\langle \textstyle\prod_{t:\Psi(t)} A(t) \Big|^{\Phi}_{a} \right\rangle \leftrightsquigarrow \mathrm{Ext} : \left(\sum_{I,\mathrm{inc}:\Phi \subseteq \Psi} \sum_{A:\left(\sum_{i:I}[\Psi(i)]\right) \to \mathrm{ty}} \prod_{i:[I]} \prod_{t:[\Phi(i)]} A(i, \mathrm{inc}(i,t)) \right) \to \mathrm{ty}$$

Normalization, Step 2: Define neutral and normal forms

- In the language of simplicial STC, assume a proposition $\P : \Omega$ witnessing that a piece of data is pure syntax.

  $\rightsquigarrow$ modalities $\bigcirc := \bigcirc_{\P}$ and $\bullet := \bullet_{\P}$ modalities projecting computability data to its syntactic, resp., semantic part
- Lift the tope universe $\mathbb{T}$ to a computability structure $\mathbb{T}^\star$ and a normalization structure $\mathbb{T}^\# \subseteq \mathbb{T}^\star$, both *aligned* over $\mathbb{T}$, *i.e.* restricting to $\mathbb{T}$ over $\P$.
- Define neutral and normal forms, inductively:

$$\begin{aligned} \mathrm{nfty} &: \{\mathcal{U} \mid \P \hookrightarrow \mathrm{ty}\} \\ \mathrm{var}, \mathrm{nf} &: \prod_{A:\mathrm{ty}} \{\mathcal{U} \mid \P \hookrightarrow A\} \\ \mathrm{ne}_{(-)} &: \prod_{\chi:\mathbb{T}^\#} \prod_{A:\mathrm{ty}} \{\mathcal{U} \mid \bullet\chi \hookrightarrow A\} \end{aligned}$$

- Have to use *stabilized* neutrals to account for cases where terms should be evaluated further, *e.g.* for $f : (x \to_A y) \equiv \left\langle \Delta^1 \to A \middle|_{[x,y]}^{\partial\Delta^1} \right\rangle$ consider $f(t)$ vs. $f(0)$.

Normalization, Step 3: Stabilized Tait Yoga

- **Classical Tait Yoga:** Construct a family of *reflection* and *reification* maps, namely, for each computability structure $C : \mathrm{ty}^\star$ maps

$$\mathrm{ne}(C) \xrightarrow{\Uparrow_C} C \xrightarrow{\Downarrow_C} \mathrm{nf}(C)$$

  which are *vertical*, *i.e.* coincide with the identity over the syntax as tracked by ¶ : Ω.
- **Stabilized Tait Yoga (Sterling, Sterling–Angiuli):** Construct a family of *stabilized reflections*, *i.e.* , for any tope $\chi : \mathbb{T}$ maps

$$\mathrm{stabne}_\chi(C) \xrightarrow{\Uparrow_C^\chi} C \xrightarrow{\Downarrow_C} \mathrm{nf}(C)$$

  such that $\Uparrow_C^\chi$ is the identity over $\bullet_¶(\chi)$.
- The *stabilized neutrals* in $\mathrm{ne}_\chi(C)$ contain explicit computability data on the locus of instability χ.
- The stabilized Tait Yoga serves to extend the universe of *computability* structures $\mathrm{ty}^\star \to \mathrm{ty}$ to a universe of *normalization* structures

$$\mathrm{ty}^\# \to \mathrm{ty}^\star \to \mathrm{ty}.$$

Interlude: Extension types, I

- We can capture the extension type former via:

$$\mathrm{Ext} : \left(\sum_{I:\mathbb{C}} \sum_{\substack{\Phi,\Psi:[I]\to\mathbb{T}\\ \iota:\prod_{i:\mathbb{C}}[\Phi(i)]\to[\Psi(i)]}} \sum_{A:\left(\prod_{i:[I]}[\Psi(i)]\right)\to\mathrm{ty}} \prod_{\substack{i:[I]\\ t:[\Phi(i)]}} A(i,\iota(i,t))\right) \to \mathrm{ty}$$

$$\lambda_{\mathrm{Ext}} : \prod_{I,\Phi,\Psi,\iota}\prod_{A,a}\left(\prod_{i:I}\prod_{t:[\Psi(i)]}\{A(i,t) \mid \varphi \hookrightarrow a\}\right) \cong \mathrm{Ext}(I,\Psi,\Phi,\iota,A,a)$$

- This gets then lifted to the level of neutral and normal forms:

$$\mathbf{Ext} : \left\{\left(\sum_{I:\mathbb{C}}\sum_{\substack{\Psi,\Phi:[I]\to\mathbb{T}\#\\ \iota:\prod_{[I]}[\Phi]\to[\Psi]}}\sum_{A:\left(\prod_{[I]}[\Psi]\right)\to\mathrm{nfty}}\left(\prod_{\substack{i:[I]\\ t:[\Phi(i)]}} \mathrm{nf}(A(i,\iota(i,t)))\right)\right)\to \mathrm{nfty} \;\middle|\; \P\hookrightarrow \mathrm{Ext}\right\}$$

$$@_{\mathrm{Ext}} : \prod_{I,\Psi,\Phi,\iota}\prod_{A,a}\prod_{\chi:\mathbb{T}\#}\left\{\mathrm{ne}_\chi(\mathrm{Ext}(I,\Psi,\Phi,\iota,A,a)) \to \prod_{i:[I]}\prod_{t:[\Psi(i)]} \mathrm{ne}_{\chi\sqcup\#\varphi(i,t)}(A(i,t)) \mid \P \hookrightarrow \lambda f,i,t.f(i,t)\right\}$$

$$\boldsymbol{\lambda}_{\mathbf{Ext}} : \prod_{I,\Psi,\Phi,\iota}\prod_{A,a}\left\{\prod_{b:\prod_{i:I}\prod_{t:[\Psi(i)]}\{A(i,t)\mid\varphi\hookrightarrow a\}} \mathrm{nf}(\mathrm{Ext}(A,\lambda i.\lambda t.b)) \mid \P\hookrightarrow \lambda f,i,t.f(i,t)\right\}$$

Interlude: Extension types, II

- Normalization structure for extension types:

$$\mathrm{Ext}^{\#} : \left\{ \left(\sum_{I:\mathbb{C}} \sum_{\substack{\Psi,\Phi:[I]\to\mathbb{T}^{\#} \\ \iota:\prod_{[I]}[\Phi]\to[\Psi]}} \sum_{A:\left(\prod_{[I]}[\Psi]\right)\to\mathrm{ty}^{\#}} \left(\prod_{\substack{i:[I] \\ t:[\Phi(i)]}} \mathrm{nf}(A(i,\iota(i,t))) \right) \right) \to \mathrm{ty}^{\#} \;\middle|\; \P \hookrightarrow \mathrm{Ext} \right\}$$

- Stable reflection and reification are then defined recursively[4] as follows:

$$\Downarrow_{\mathrm{ty}} \mathrm{Ext}^{\#}(A,a) := \mathbf{Ext}(\lambda i,t.\Downarrow_{\mathrm{ty}} A(i,t), \lambda i.\Downarrow_{A(i)} a(i))$$

$$\Uparrow^{\chi}_{\mathrm{Ext}^{\#}(A,a)} (f) := \lambda i,t.\Uparrow^{\chi\sqcup^{\#}\varphi(i,t)}_{\mathrm{Ext}^{\#}(A,a)} [@_{\mathrm{Ext}}(f_0,i,t) \mid \chi\sqcup^{\#}\varphi(i,t) \hookrightarrow [\chi \hookrightarrow f(i,t), \varphi(i,t) \hookrightarrow a(i,t)]]$$

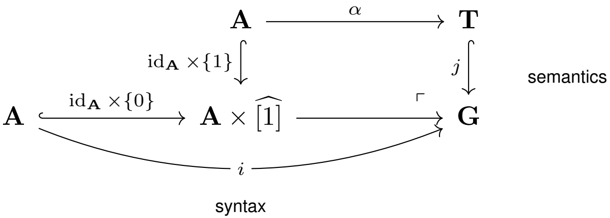
$$\Downarrow_{\mathrm{Ext}^{\#}(A,a)} (f) := \lambda_{\mathbf{Ext}}(\lambda i,t.\Downarrow_{A(i,t)} (f(i,t)))$$

---

[4] compare with the classical definitions for function types

Normalization, Step 4: Computability topos & normalization functions

- Denote by $\mathcal{T}$ the category of judgments of STT and by $\mathcal{A}$ the category of *atomic contexts*. The morphisms in each case are renamings of variables.
- Canonical figure shape induces geometric morphism on presheaf toposes: $\alpha : \mathcal{A} \to \mathcal{T} \rightsquigarrow \alpha : \widehat{\mathcal{A}} =: \mathbf{A} \to \mathbf{T} := \widehat{\mathcal{T}}$
- The computability topos $\mathbf{G}$ arises as pushout:



- From $j : \mathbf{T} \to \mathbf{G}$, we can extract the syntactic open $\P : \Omega$.
- In fact, $\mathbf{G}$ arises through *Artin gluing* as $\mathbf{G} \cong \mathbf{A} \downarrow \alpha^*$, where $\alpha^* \dashv \alpha$.
- Then $\mathbf{G}$ can be exhibited as a model for simplicial STC, and using its semantic description as well as the previously defined internal normalization structure one can give an internal construction of the desired normalization functions.

Current and future work

- Todo: Complete construction of simplicial STC framework and proofs.
- Desired base types as well as Martin-Löf intensional identity types still missing.
- Envisioning future use for even more complex theories such as *(multi-)modal simplicial type theories*, in a modular way.

References

- T. Altenkirch, M. Hofmann, T. Streicher (1995): Categorical reconstruction of a reduction free normalization proof. Category Theory and Computer Science. doi:10.1145/571157.571161
- U. Buchholtz, J. Weinberger (2021): Synthetic Fibered $(\infty, 1)$-Category Theory Preprint arXiv:2105.01724
- M. Fiore (2002): Semantic Analysis of Normalisation by Evaluation for Typed Lambda Calculus. Proc of ACM SIGPLAN PPDP '02. doi:10.1145/571157.571161
- E. Riehl, M. Shulman (2017): A Type Theory for Synthetic ∞-Categories Higher Structures **1**, no. 1, 147-–224.
- J. Sterling (2021): First Steps in Synthetic Tait Computability: The Objective Metatheory of Cubical Type Theory. Doctoral Dissertation, Carnegie Mellon University
- J. Sterling, C. Angiuli (2021): Normalization for Cubical Type theory. 36th Annual Symposium on Logic in Computer Science (LICS). doi:10.1109/LICS52264.2021.9470719
- W. W. Tait (1967): Intensional Interpretations of Functionals of Finite Type I. The Journal of Symbolic Logic 32.2 (1967), pp. 198–212.
- J. Weinberger (2022): Strict stability of extension types. Preprint arXiv:2203.07194